In [1]:
```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import warnings
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
```

In [2]:
```
warnings.filterwarnings('ignore')
```

In [3]:
```
# Creating a synthetic dataset
# This dataset simulates customer data for a telecom company
data = {
      'CustomerID': range(1, 101),  # Unique ID for each customer
      'Age': [20, 25, 30, 35, 40, 45, 50, 55, 60, 65]*10,  # Age of customer
      'MonthlyCharge': [50, 60, 70, 80, 90, 100, 110, 120, 130, 140]*10,  #
      'CustomerServiceCalls': [1, 2, 3, 4, 0, 1, 2, 3, 4, 0]*10,  # Number c
      'Churn': ['No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Y
}
df = pd.DataFrame(data)
```

In [4]:
```
# Splitting the dataset into features and target variable
# Features include age, monthly charge, and customer service calls
# The target variable is churn (Yes or No)
X = df[['Age', 'MonthlyCharge', 'CustomerServiceCalls']]
y = df['Churn']
```

In [5]:
```
# Splitting the dataset into training and testing sets
# 70% of the data is used for training and 30% for testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, ran
```

In [6]:
```
# Training the Decision Tree model
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

Out[6]:

▾ DecisionTreeClassifier

▶ Parameters

In [7]:
```
# Making predictions on the test set
y_pred = clf.predict(X_test)
```

In [8]:
```
# Evaluating the model using accuracy
# Accuracy is the proportion of correct predictions among the total number c
accuracy = accuracy_score(y_test, y_pred)
print(f'Model Accuracy: {accuracy}')
```

```
Model Accuracy: 1.0
```

In [9]:
```
# Visualizing the decision tree
# This visualization helps in understanding how the model makes decisions
plt.figure(figsize=(12,8))
tree.plot_tree(clf, filled=True, feature_names=['Age', 'MonthlyCharge', 'Cus
plt.title('Decision Tree for Predicting Customer Churn')
plt.show()
```